Service的概念

（1）Service是一种Android的组件，可以在后台长时间运行

（2）Service不提供界面交互

即使用户跳转至另一个应用后，Service仍旧在后台运行

任意应用组件都可以绑定服务，甚至可以用来完成进程间通讯的任务

eg：

需要下载时

播放音乐

文件I/O

Service的启动方式

Service可以通过两种方式来调用

start

一旦某个组件start一个Service后，Service开始独立运行，在它自己独立的一个线程中运行，不再与原来的组件产生任何关系

bind

某个组件bind一个Service后，Service为组件提供一个接口，近似与客户端进行连接，会进行互相的交互，和Activity运行在同一个进程中，但仍然是两个独立的线程。

实例1：

AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.amaker.hjs.servicedemo.service"**>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".DemoActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
  
 <**service android:name=".MyService"**>  
 <**intent-filter**>  
 <**action android:name="heying.myService"**/>  
 </**intent-filter**>  
 </**service**>  
  
 <**service android:name=".ComputeService"**>  
 <**intent-filter**>  
 <**action android:name="heying.computeService"**></**action**>  
 </**intent-filter**>  
 </**service**>  
 </**application**>  
  
</**manifest**>

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
  
 <**Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="28dp"  
 android:layout\_marginTop="42dp"  
 android:text="startService"** />  
  
 <**Button  
 android:id="@+id/btnStop"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBottom="@+id/btnStart"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="93dp"  
 android:text="stopService"** />  
  
 <**Button  
 android:id="@+id/btnBind"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignStart="@+id/btnStart"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="142dp"  
 android:text="BindService"** />  
  
 <**Button  
 android:id="@+id/btnUnBind"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignStart="@+id/btnStop"  
 android:layout\_alignBottom="@+id/btnBind"  
 android:text="UnBindService"** />  
</**RelativeLayout**>

MyService.java

**package** com.amaker.hjs.servicedemo.service;  
  
**import** android.app.Service;  
**import** android.content.Intent;  
**import** android.os.Binder;  
**import** android.os.IBinder;  
**import** android.util.Log;  
  
**public class** MyService **extends** Service {  
 **private static final** String ***TAG***=**"MyService"**;  
  
 */\*\*  
 \* 只有生命周期的第一次才调用  
 \*/* @Override  
 **public void** onCreate() {  
 Log.*i*(***TAG***,**"onCreate"**);  
 **super**.onCreate();  
 }  
  
  
 */\*\*  
 \* 每一次startService启动都会被调用  
 \** ***@param intent*** *\** ***@param flags*** *\** ***@param startId*** *\** ***@return*** *\*/* @Override  
 **public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  
 Log.*i*(***TAG***,**"onStartCommand"**);  
 **return super**.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 **public void** onDestroy() {  
 Log.*i*(***TAG***,**"onDestroy"**);  
 **super**.onDestroy();  
 }  
  
  
 */\*\*  
 \* bindService启动时调用，在整个生命周期中只被调用一次，必须有IBinder对象返回，才表示连接成功，  
 \* ，才会自动调用与Service的建立的连接——ServiceConnection接口实例conn的两个回调方法  
 \** ***@param intent*** *\** ***@return*** *\*/* @Override  
 **public** IBinder onBind(Intent intent) {  
 Log.*i*(***TAG***,**"onBind"**);  
 **return new** MyBinder();  
 }  
  
 **class** MyBinder **extends** Binder { }  
}

MainActivity.java

**package** com.amaker.hjs.servicedemo.service;  
  
**import** android.app.Activity;  
**import** android.content.ComponentName;  
**import** android.content.Intent;  
**import** android.content.ServiceConnection;  
**import** android.os.Bundle;  
**import** android.os.IBinder;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
  
**public class** MainActivity **extends** Activity **implements** View.OnClickListener {  
 **private** Button **btnStart**,**btnStop**;  
 **private** Button **btnBind**,**btnUnBind**;  
  
 **private** ServiceConnection **conn** = **new** ServiceConnection() {  
 *//这里必须实现接口的两个回调方法* @Override  
 **public void** onServiceConnected(ComponentName name, IBinder service) {  
 *//连接成功，自动调用* Log.*i*(**"Activity"**,**"onServiceConnected"**);  
 }  
  
 @Override  
 **public void** onServiceDisconnected(ComponentName name) {  
 *//如果出现问题，连接无效时调用* Log.*i*(**"Activity"**,**"onServiceDisconnected"**);  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 init();  
 }  
  
 **private void** init(){  
 **btnStart** = (Button)findViewById(R.id.***btnStart***);  
 **btnStop**=(Button)findViewById(R.id.***btnStop***);  
  
 **btnStart**.setOnClickListener(**this**);  
 **btnStop**.setOnClickListener(**this**);  
  
  
 **btnBind**=(Button)findViewById(R.id.***btnBind***);  
 **btnUnBind**=(Button)findViewById(R.id.***btnStop***);  
  
 **btnBind**.setOnClickListener(**this**);  
 **btnUnBind**.setOnClickListener(**this**);  
  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **null**;  
 **switch** (v.getId()){  
 **case** R.id.***btnStart***:*//启动MyService服务（组件的启动必须在AndroidManifest.xml进行声明）  
 //intent = new Intent("heying.myService");//隐式声明 放入定义好的逻辑动作名，这里只能用intent显式说明* intent=**new** Intent(**this**,MyService.**class**); *//Android 5.0以上必须用显式声明* startService(intent); *//onCreate->onStartCommand* **break**;  
 **case** R.id.***btnStop***:  
 *//intent = new Intent("heying.myService");//放入定义好的逻辑动作名* intent = **new** Intent(**this**,MyService.**class**);  
 stopService(intent);  
 **break**;  
 **case** R.id.***btnBind***:  
 intent=**new** Intent(**this**,MyService.**class**);*//Android 5.0以上必须用显式声明* bindService(intent,**conn**,***BIND\_AUTO\_CREATE***);  
 *//第一个参数是关联好Service的intent；  
 // 第二个参数是和Service建立的连接，即客户端和服务器端必须要有一个连接的存在；  
 // 第三个参数是flag，表示如果连接的过程中MyService不存在，那么系统就自动创建一个Service实例  
 //onCreate->onBind->onServiceConnected* **break**;  
 **case** R.id.***btnUnBind***:  
 intent = **new** Intent(**this**,MyService.**class**);  
 unbindService(**conn**);  
 *//取消绑定只要把连接关掉就可以了* **break**;  
 }  
 }  
}

错误整理：

在AndroidManifest.xml中声明Service组件：

<**service android:name=".MyService"**>  
 <**intent-filter**>  
 <**action android:name="heying.myService"**/>  
 </**intent-filter**>  
  
</**service**>

在MainActivity.java中进行隐式Intent声明如下：

intent = **new** Intent(**"heying.myService"**);

startService(intent);

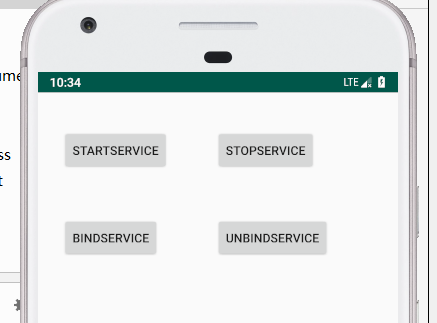
会报出如下错误：

java.lang.IllegalArgumentException: Service Intent must be explicit: Intent {act=heying.myService }

这是由于在Android5.0中service的intent必须要进行显性声明，绑定时才不会报错

intent=**new** Intent(**this**,MyService.**class**);

startService(intent);



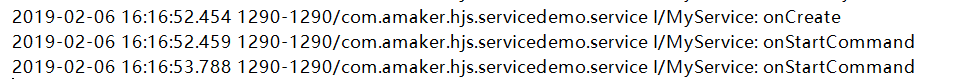
点击startService按钮



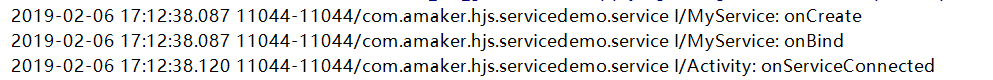
点击stopService按钮



此时再连续点击两次startService按钮，第二次只调用了onStartCommand



点击bindService按钮



再次点击bindService按钮没有任何现象产生

实例2：

activity\_demo.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="50dp"**>  
 <**ImageView  
 android:layout\_width="110dp"  
 android:layout\_height="110dp"  
 android:src="@drawable/hdu"  
 android:layout\_gravity="center\_horizontal"**/>  
  
 <**TextView  
 android:layout\_marginTop="20dp"  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:gravity="center\_horizontal"  
 android:text="学生成绩统计"  
 android:textColor="#F00"  
 android:textSize="30dp"  
 android:textStyle="bold"** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="30dp"**>  
  
 <**TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="操作系统："  
 android:textSize="20dp"  
 android:gravity="center"**/>  
  
 <**EditText  
 android:id="@+id/etOs"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:text=""** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="编译原理："  
 android:textSize="20dp"  
 android:gravity="center"**/>  
  
 <**EditText  
 android:id="@+id/etCp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:text=""** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="数据结构："  
 android:textSize="20dp"  
 android:gravity="center"**/>  
  
 <**EditText  
 android:id="@+id/etDs"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:text=""** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.6"  
 android:text="计算机网络："  
 android:textSize="20dp"  
 android:gravity="center"**/>  
  
 <**EditText  
 android:id="@+id/etCn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:text=""** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**Button  
 android:id="@+id/btnConn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginTop="30dp"  
 android:text="建立Service连接"** />  
  
 <**Button  
 android:id="@+id/btnDisplay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginTop="30dp"  
 android:text="计算并显示平均成绩"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**Button  
 android:id="@+id/btnTest"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="startService测试"  
 android:layout\_marginTop="10dp"  
 android:layout\_gravity="center\_horizontal"  
 android:onClick="test"**/>  
  
 </**LinearLayout**>  
  
  
 <**TextView  
 android:id="@+id/tvResult"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="您的成绩"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="20dp"  
 android:textSize="20dp"  
 android:textColor="#F00"**/>  
  
</**LinearLayout**>

ComputeService.java

**package** com.amaker.hjs.servicedemo.service;  
  
**import** android.app.Service;  
**import** android.content.Intent;  
**import** android.os.Binder;  
**import** android.os.IBinder;  
**import** android.util.Log;  
  
**public class** ComputeService **extends** Service {  
  
  
 *//（2）startService方式：只有接收来自Activity的请求，Service并不能返回值* @Override  
 **public int** onStartCommand(Intent intent, **int** flags, **int** startId) {  
 **double** os = intent.getDoubleExtra(**"os"**,0);  
 **double** cp = intent.getDoubleExtra(**"cp"**,0);  
 **double** ds = intent.getDoubleExtra(**"ds"**,0);  
 **double** cn = intent.getDoubleExtra(**"cn"**,0);  
  
 **double** result = inServiceCal(os,cp,ds,cn);  
 Log.*i*(**"Test"**,result+**""**);  
 **return super**.onStartCommand(intent, flags, startId);  
 }  
  
  
  
  
 *//（1）BinderService 方式：对于接收来自Activity的请求，Service有返回值（Binder）* @Override  
 **public** IBinder onBind(Intent intent) {  
 **return new** ComputeBinder();  
 }  
  
 **public class** ComputeBinder **extends** Binder{  
  
 *//外部类对象不能调用内部类对象的方法，所以必须把计算方法写到外面* **public double** calcAvg(**double**...scores){  
 **return** inServiceCal(scores);  
 }  
 }  
  
  
  
 */\*\*  
 \* 计算平均成绩  
 \** ***@param scores*** *可变数组，参数可以是0-N个  
 \** ***@return*** *\*/* **public double** inServiceCal(**double**...scores){  
 **int** count = scores.**length**;  
 **if** (count==0){  
 **return** 0;  
 }  
 **double** sum = 0;  
 **for** (**double** s:scores){  
 sum+=s;  
 }  
 **return** sum/count;  
  
 }  
}

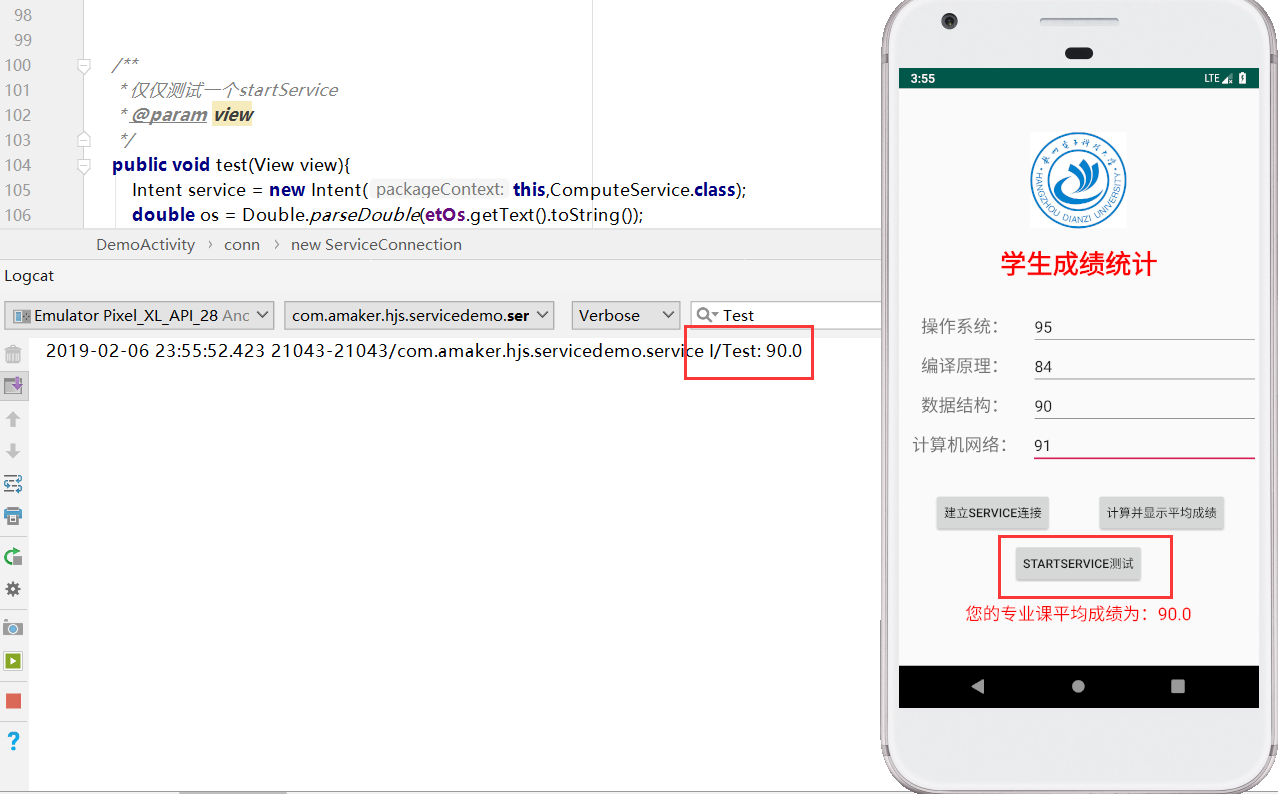
DemoActivity.java

**package** com.amaker.hjs.servicedemo.service;  
  
**import** android.app.Activity;  
**import** android.content.ComponentName;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.ServiceConnection;  
**import** android.os.Bundle;  
**import** android.os.IBinder;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
  
**public class** DemoActivity **extends** Activity {  
 **private** EditText **etOs**, **etCp**, **etDs**, **etCn**;  
 **private** TextView **tvResult**;  
 **private** Button **btnConn**, **btnDisplay**;  
  
 **private** ComputeService.ComputeBinder **binder** = **null**;  
  
 **private** ServiceConnection **conn** = **new** ServiceConnection() {  
  
 *//连接成功* @Override  
 **public void** onServiceConnected(ComponentName name, IBinder service) {  
 **binder** = (ComputeService.ComputeBinder) service;  
  
 }  
  
 @Override  
 **public void** onServiceDisconnected(ComponentName name) {  
  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_demo***);  
  
 *//最好将启动服务写到onCreate()方法里* Intent service = **new** Intent(DemoActivity.**this**,ComputeService.**class**);  
 bindService(service, **conn**, Context.***BIND\_AUTO\_CREATE***);  
  
  
 init();  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 unbindService(**conn**);*//在关闭Activity时必须将Service也关掉，将服务回收* **super**.onDestroy();  
 }  
  
 **private void** init() {  
 **etOs** = (EditText) findViewById(R.id.***etOs***);  
 **etCp** = (EditText) findViewById(R.id.***etCp***);  
 **etDs** = (EditText) findViewById(R.id.***etDs***);  
 **etCn** = (EditText) findViewById(R.id.***etCn***);  
 **tvResult** = (TextView) findViewById(R.id.***tvResult***);  
 **btnConn** = (Button) findViewById(R.id.***btnConn***);  
 **btnDisplay** = (Button) findViewById(R.id.***btnDisplay***);  
  
 **btnConn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//启动Service* Intent service = **new** Intent(DemoActivity.**this**,ComputeService.**class**);  
  
 bindService(service, **conn**, Context.***BIND\_AUTO\_CREATE***);  
 *//注意，从启动服务开始，与获得连接后返回的Binder对象，这是一个异步的过程，即获得Binder对象是需要一定时间的  
 // 因此接下来这个方法里的代码不能有任何关于操作返回的Binder对象的代码，不然会报空指针异常* }  
 });  
  
 **btnDisplay**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **try** {  
 *//从文本框中读取所输入的成绩* **double** os = Double.*parseDouble*(**etOs**.getText().toString());  
 **double** cp = Double.*parseDouble*(**etCp**.getText().toString());  
 **double** ds = Double.*parseDouble*(**etDs**.getText().toString());  
 **double** cn = Double.*parseDouble*(**etCn**.getText().toString());  
 *//请求对象完成工作* **if** (**binder**!=**null**) {  
 **double** result = **binder**.calcAvg(os, cp, ds, cn);  
 *//显示* **tvResult**.setText(**"您的专业课平均成绩为："** + result);  
 }  
 } **catch** (NumberFormatException ex) {  
 ex.printStackTrace();  
 }  
 }  
 });  
 }  
  
  
 */\*\*  
 \* 仅仅测试一个startService  
 \** ***@param view*** *\*/* **public void** test(View view){  
 Intent service = **new** Intent(**this**,ComputeService.**class**);  
 **double** os = Double.*parseDouble*(**etOs**.getText().toString());  
 **double** cp = Double.*parseDouble*(**etCp**.getText().toString());  
 **double** ds = Double.*parseDouble*(**etDs**.getText().toString());  
 **double** cn = Double.*parseDouble*(**etCn**.getText().toString());  
 service.putExtra(**"os"**,os);  
 service.putExtra(**"cp"**,cp);  
 service.putExtra(**"ds"**,ds);  
 service.putExtra(**"cn"**,cn);  
 startService(service);  
  
 }  
  
  
}  
  


点击“计算并显示平均成绩”（BinderService，有返回）



点击“StartService测试”（startService方式，只能传值，无返回）



注：“建立Service连接”按钮，由于改写创建连接的方法在DemoActivity的onCreate方法里，所以有无点击均已有Binder对象的创建。